

订餐系统数据库设计与实现

杨坤鹏 曾 丽

成都锦城学院计算机与软件学院 四川 成都 611731

【摘要】在现在的快节奏生活中，人们对于网上点餐订餐的需求也在不断提高。本文主要从需求分析，数据库设计分析以及数据库操作模拟等三个方面阐述了订餐系统的数据库设计与实现，本系统实现了用户、菜品、餐饮机构、订单、配送员等数据的管理，具有使用方便，能及时、灵活地对所需数据进行查询、修改、删除等操作等优点。

【关键词】订餐系统；数据库；SQL 语言

1 导论

1.1 设计背景与意义

近几年来，随着人民生活水平的不断提高，人们日常生活的节奏越来越快，人们对餐饮的要求也随之提高，不少人在要求色香味俱全的同时，也对与就餐的时间、地点和方式有了更高的要求。然而，传统餐饮企业依旧存在由于自身地理位置、客人流动性大以及部分距离餐厅位置较近的顾客由于工作或学业的繁忙，无法抽出时间来享受美味等问题而导致的客户流失现象，为解决这样的问题，便产生了顾客快捷订餐的需求。在其中，最便利的方式莫过于将现如今我们发达的互联网和餐饮行业结合起来，形成一种网上订餐系统，这样不仅会提高餐饮行业的服务质量，让其多一分销售市场，而且为用户提供了最方便的食品服务方法，可使很多美食爱好者足不出户就可以享受到各式各样的美食。

2 需求分析

2.1 数据需求分析

作为一款订餐系统数据库，面向的主要对象就是我们的用户、配送员、管理员和餐饮机构负责人，为此需要以订单为中心，了解有关于用户、配送员、菜品和餐饮机构的全部信息。

2.2 功能需求分析

本次的订餐系统数据库设计主要是针对订单信息的管理，例如：对订单的详细信息查询、订单的内容更新、订单的取消，最终达到一个便于使用，易于操作的目的。对于管理者，需要给予他所有的权限，管理者可以操作整个数据库并且可以对整个数据库进行在规则内的随意操作；对于用户，需要让用户只可以看到属于他自己的信息相关的订单信息以及订单信息中的各项内容等，还需允许用户自由进行评价、取消订单、点餐等操作；对于配送员，需要给予他管理自己个人信息，以及进行注册注销、接单等相关操作的权利；对于餐饮机构负责人，需要给予他修改店名、联系方式、接受订单等权利，需要记录他的位置信息，联系方式、出餐时间和接单时间等信息。对于订单，需要记录订单中用户的点餐数量，下单时间，商家接单时间，商家出餐时间，配送员接单时间，菜品送达时间，订单完成时间，用户评价等信息，可以说，订单是联系起整个系统的核心。

3 数据库设计

3.1 系统 E—R 图设计

本次订餐系统经过需求分析，我们了解到具体的实体信息分别是用户实体，配送员实体，订单实体，菜品实体，餐饮机构实体。据需求分析的结果，由用户，配送员，订单，餐饮机构，菜品设计出系统总的 E—R 图，见图一：

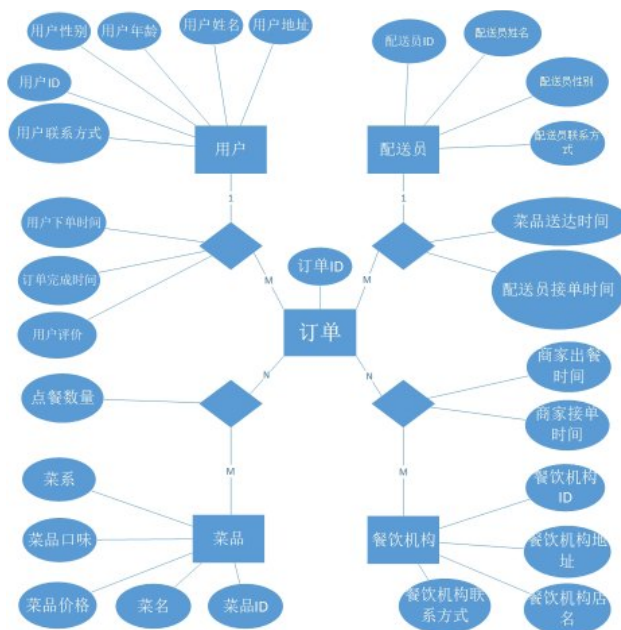


图 1 订餐系统 ER 图

3.2 逻辑结构设计

通过 E-R 图我们可以得知不同实体之间的联系，通过 E-R 模型我们可以转换为关系模型，得到的关系模式如下：

用户（用户 ID，用户性别，用户年龄，用户姓名，用户地址，用户联系方式）

配送员（配送员 ID，配送员姓名，配送员性别，配送员联系方式）

订单（订单 ID，配送员 ID，用户 ID，餐饮机构 ID，菜品 ID，点餐数量，用户下单时间，商家接单时间，商家出餐时间，配送员接单时间，菜品送达时间，订单完成时间，用户评价）

菜品 (菜系, 口味, 价格, 菜名, 菜品 ID)

餐饮机构 (餐饮机构 ID, 餐饮机构地址, 餐饮机构店名, 餐饮机构电话)

菜品_订单 (订单 ID, 菜品 ID, 点餐数量)

餐饮机构_订单 (订单 ID, 餐饮机构 ID, 点餐数量)

4 订餐系统数据库实现

4.1 基本表的创建

在这里, 我们一共建立了菜品、菜品_订单、餐饮机构、餐饮机构_订单、订单、配送员、用户等七张表格用于存储本系统的相关数据, 现在列出部分表格如下:

订单信息表是整个订餐系统数据库的核心。没有订单, 就没有订餐系统的正常运行, 所有的实体都会与订单发生一定的联系, 是订餐系统的中流砥柱。

```
create table 订单
```

```
( 订单 ID int primary key identity, 配送员 ID int foreign key(配送员 ID) references 配送员 (配送员 ID), 用户 ID int foreign key(用户 ID) references 用户 (用户 ID), 用户下单时间 datetime, 配送员接单时间 datetime, 菜品送达时间 datetime, 订单完成时间 datetime, 用户评价 char(100))
```

由于一个菜品可以被多个用户通过多个订单进行点餐, 一个订单可以包含多种或多个菜品, 餐饮机构与订单的联系也是如此, 所以需要建立一个菜品_订单多对多联系表, 相应的, 餐饮机构_订单表也属于多对多关系产生的表。

```
create table 菜品_订单
```

```
( 订单 ID int, foreign key(订单 ID) references 订单 (订单 ID), 菜品 ID int, foreign key(菜品 ID) references 菜品 (菜品 ID), 点餐数量 int not null)
```

4.2 存储过程的建立

在本系统中, 有着各种的业务需求需要完成, 比如商家添加菜品、修改菜品价格, 用户注册注销、修改个人信息、投诉配送员以及评价等, 这些需要完成的动作, 在这里我都将他们设计成了各种的存储过程, 现将展示几个存储过程:

当用户对自己收到的商品不满意时, 查询到配送员的相关信息, 之后便可以对配送员进行投诉和评价。

```
create proc 用户投诉配送员和评价 @cname char(20), @pname char(20), @appraise char(100) as
declare @csex char(1), @cage int, @cad char(20), @cphone char(20), @cid int, @psex char(1), @pphone char(20), @pid int
```

```
select @csex=用户性别, @cad=用户地址, @cphone=用户联系方式, @cage=用户年龄, @cid=用户 ID from 用户 where 用户姓名=@cname
```

```
select @psex=配送员性别, @pphone=配送员联系方式, @pid=配送员 ID from 配送员 where 配送员姓名=@pname
```

```
update 订单 set 用户评价=@appraise from 订单 where 用户 ID=@cid print '已评价'
```

当用户进行点餐操作时, 在订单中插入一条链接菜品和订单的数据, 并根据用户所点菜品的单价和数量得出其所要支付的总金额。

```
create proc 提示应支付金额 @菜名 char(20), @点餐数量 int as
declare @菜品价格 int
select @菜品价格=菜品价格 from 菜品 where 菜名=@菜名
print '应支付金额为: ' + @菜品价格 * @点餐数量
```

4.3 触发器的建立

当某用户进行点餐操作时, 会在菜品信息表与订单信息表中起链接作用的菜品_订单表中生成一个由菜品 ID、订单 ID 和点餐数量组成的数据, 为防止铺张浪费, 我们通过这个后触发器向用户进行提示, 即确认自己的点餐数量大于三。

```
create trigger 点餐提示 on 菜品_订单 after insert as
if (select count(*) from 菜品_订单 where 菜品 ID in (select 菜品 ID from inserted))>=3
print '请再次确认您的订单'
```

用户下单后, 会在订单信息中插入一条有关于用户下单时间、用户 ID、订单 ID、配送员接单时间、菜品送达时间、订单完成时间和用户评价的新数据。根据用户下单时间, 当配送员在接单半个小时之后, 如果仍未插入订单完成时间, 则给配送员一个提示: “距离接单时间已超过半小时, 请尽快完成订单”。

```
create trigger 提示送餐 on 订单 after insert as
if ((select 配送员接单时间 from 订单)+30<(select 订单完成时间 from 订单))
print '距离接单时间已超过半小时, 请尽快完成订单'
```

5 数据库测试

在我们对数据库的基本设计结束之后, 我们需要对数据库整体的情况进行测试, 通过模拟实际情况来运行数据库应用程序, 执行各种操作, 以便观察数据库是否达到了我们的预期, 如果不满足, 则需要对应用程序进行修改与调整, 直至数据库达到设计要求。整个阶段在本文中主要分为数据库数据载入测试和相关操作功能测试。

5.1 输入数据测试

增加新的用户信息、菜品信息、配送员信息、餐饮机构信息、订单信息、菜品_订单信息和餐饮机构_订单信息, 运行其中相关的检测用户注册、商家添加菜品等由存储过程实现的功能。

```
insert into 用户 values ('X', 20, 'XXX', 'XXX', 'B003')
insert into 菜品 values ('XX', 'XXXX', 20, '宫保鸡丁')
insert into 配送员 values ('XXX', 'X', '18739380056')
```

5.2 完成相关操作测试

现模拟进行操作的情景, 相关操作如下:

某个用户想要看到销量排行前三的菜品以及各自的点餐人数。首先, 菜品的点餐数量是存储在表格菜品_

订单中的数据,那么我们只需要将表格中的数据根据菜品 ID 进行组合和倒叙排序,再进行检索就可以得出相关数据。

```
select top 3 菜品 ID, count(菜品 ID), sum(点餐数量) from 菜品_订单
```

```
group by 菜品 ID order by sum(点餐数量) DESC
```

当管理员想要查询点过编号为 1 的菜品的用户编号和姓名时,可以使用多表连接,将用户表、订单表、菜品_订单表连接起来,这样便可以查询出点过这款菜品的用户的各种信息,本操作中只展示了用户 ID 和姓名

```
select Y. 用户 ID,Y. 用户姓名 FROM 用户 Y join
```

```
订单 D ON D. 用户 ID=Y. 用户 ID
```

```
JOIN 菜品_订单 CD ON D. 订单 ID= CD. 订单 ID
```

```
where CD. 菜品 ID=1
```

6 系统安全与维护

在我们进行信息管理的过程中,需要通过数据库系统来实现大量的数据的存储、共享、访问以及修改,因此科学有效的进对我们所设计的数据库系统进行管理与维护,保证数据库中数据的安全性、完整性和有效性,也成为了我们必须要考虑的一环。对数据库的经常性的维护工作,主要由数据库管理员来完成。

对于本系统的安全来讲,我们需要通过建立面向不同对象的视图来限制其在整个数据库中所能查看的内容,以及通过各种索引来加快相关数据的搜索速度和准确性,这样便可在一定程度上保障数据的安全性、查询快捷性与系统友好性。

对于本系统的维护方面,主要进行有关于完整性控制与安全性方面的监督、分析、改造和维护。对于系统的维护,包括备份数据库数据、恢复数据库系统、建立

各种对象信息表、见识系统运行状况以及保证最基本的系统数据的安全。事实上,DBA 定期地对整个数据库系统进行的监督、分析、改造与维护,不仅可以确保数据库系统的正常运行,而且可以根据自身的市场需求对数据库进行一定程度上的功能拓展,增加数据库系统的灵活性。

7 数据库开发总结

本文是以当今社会下的互联网技术发展和传统餐饮行业的弊端为大背景设计了一款由 T-SQL 语言编写而成的网上订餐系统数据库。在本次设计中,主要从结构设计和行为设计出发,先是进行了整个模型的逻辑设计与梳理,后运用了软件工程相关的 ER 图模型设计,最后通过建立各种实体以及相关联系表、存储过程、触发器等来对数据库的功能进行基本的实现。本系统能够在满足人们对于网上订餐的基本需求上,实现通过 DBA 的对用户、菜品、餐饮机构、订单以及配送员等数据的管理,具有使用便利,能够有效及时、灵活准确地对所需数据进行增、删、查、改等操作的优点。

【参考文献】

- [1] 何玉洁. 数据库系统教程 [M]. 北京:人民邮电出版社,2015:78-105.
- [2] 王珊,萨师煊. 数据库系统概论(第 5 版) [M]. 北京:高等教育出版社,2006:76-92.